May 1968

RESEARCH ANALYSIS CORPORATION

On Variable Metric Methods of Minimization

AD 666 700

PRO CON RAC

Best Available Copy

ADVANCED RESEARCH DEPARTMENT
TECHNICAL PAPER RAC-TP-302
Published February 1968

On Variable Metric Methods of Minimization

by
John D. Pearson

May 1968 edition

RESEARCH ANALYSIS CORPORATION
McLEAN, VIRGINIA

Published February 1968
by
RESEARCH ANALYSIS CORPORATION
McLean, Virginia 22101

FOREWORD

This paper examines a class of variable metric methods of minimizing unconstrained functions that arise when the Sequential Unconstrained Minimization Technique (SUMT) is applied to general nonlinear programming problems. The methods considered require a knowledge of only the first derivatives of the function to be minimized but proceed to estimate the inverse hessian of second partial derivatives during the course of a series of one-dimensional minimizations.

Three new algorithms and the Fletcher-Powell-Davidon algorithm are derived using simple properties of a general solution to the problem of estimating the inverse hessian. Results of numerical calculations for several examples show the relative merits of the new algorithms compared to several in current use.

Nicholas M. Smith
Head, Advanced Research Department

ACKNOWLEDGMENTS

The availability of the SUMT programming system, numerous test problems, and theoretical results is due to the efforts of Garth P. McCormick, Anthony V. Fiacco, and W. Charles Mylander.

CONTENTS

Tables

On Variable Metric
Methods of Minimization

ABSTRACT

Two basic approaches to the generation of conjugate directions are considered for the problem of unconstrained minimization of quadratic functions. The first approach results in a projected gradient algorithm that gives "n step" convergence for a quadratic. The second approach is based on the generalized solution of a set of undetermined linear equations, various forms of which generate various new algorithms all giving n-step convergence. One of them is the Fletcher and Powell modification of Davidon's method.

Results of an extensive numerical comparison of these methods with the Newton-Raphson method and Fletcher-Reeves method are included.

1. INTRODUCTION

General

Let A be an $n \times n$ positive definite symmetric matrix; let b be an arbitrary n vector and c an arbitrary constant.

Consider the problem of finding the minimizing n vector $x = x^*$, for a quadratic function $f(x)$ defined by

$$f(x) = \tfrac{1}{2}x'Ax + b'x + c \tag{1}$$

The methods considered here, called variously "variable metric,"[1] "quasi-Newton,"[2,3] or "large-step gradient methods,"[4] consist of selecting an $n \times n$ matrix H_i at stage i and forming the direction $d_i = H_i' g_i$ where g_i is the gradient of $f(x)$ at x_i. A step of length α_i is chosen so that $x_i + \alpha_i d_i$ is the minimum of $f(x_i + \alpha_i d_i)$, i.e., where $d_i' g_{i+1} = 0$. H_i is then updated using $(x_{i+1} - x_i)$ and $(g_{i+1} - g_i)$. If $H_i = I$ this is the method of steepest descent. The Newton-Raphson method is obtained with $H_i = A^{-1}$. wever, on a nonquadratic function A or its equivalent the hessian of $f(x)$ may not be available. It is then of general interest to examine methods that utilize only first-derivative information and that in addition may estimate A.

Since, as is reviewed in Sec 2, a quadratic can be minimized in n steps if d_0, d_1, d_{n-1} are conjugate directions, this paper studies a class of H_i matrices that will generate conjugate directions. In Sec 3, H_i is chosen as a projection matrix and, in Sec 4, H_i is chosen as a solution to the equation $H_i Y_i = S_i$. The Fletcher-Powell-Davidon[5] algorithm is shown to be a member of this latter class. A numerical comparison of several new algorithms with the Fletcher-Powell-Davidon and the Fletcher-Reeves algorithm is given in Sec 5.

Notation

At iteration i the following column vectors occur:

- x_i is the current solution.
- g_i is the gradient of $f(x)$ at x_i.
- H_i is the current direction matrix or metric.
- d_i is the search direction from x_i, $d_i'd_i = 1$.
- $s_i = x_{i+1} - x_i = \alpha_i d_i$ is the step in x_i.
- $y_i = g_{i+1} - g_i = As_i$ is the step in g_i.
- α_i is the step length, a negative scalar.
- g_i' denotes g_i transpose, a row vector.
- $S_i = [s_0, s_1, \ldots, s_{i-1}]$ denotes a matrix with columns $s_0, \ldots, s_{i-1}$ and also without ambiguity $[s_0, s_1, \ldots, s_{i-1}]$ denotes the subspace spanned by vectors $s_0, s_1, \ldots, s_{i-1}$.
- $Y_i = [y_0, y_1, \ldots, y_{i-1}]$ denotes an $n \times i$ matrix with columns y_j.

2. PROPERTIES OF CONJUGATE DIRECTIONS

It is convenient to isolate the properties of conjugacy from the problem of generating conjugate directions as discussed in later sections.

Definition

A set of n independent directions $d_0, d_1, \ldots, d_{n-1}$ are conjugate with respect to a positive-definite symmetric matrix A if[6]

$$\begin{aligned} d_i'Ad_j &= 0 \qquad 0 \le i \ne j \le n-1 \\ d_i'Ad_i &> 0 \qquad 0 \le i \le n-1 \end{aligned} \tag{2}$$

Any point $x \in E^n$ can be represented in terms of $d_0, \ldots, d_{n-1}$ as follows: Let

$$x = \sum_{i=0}^{n-1} \lambda_i d_i$$

then

$$\lambda_i = x'Ad_i / d_i'Ad_i$$

Similarly the quadratic $f(x) = \frac{1}{2} x'Ax + x'b + c$ can be decomposed into n independent terms.

$$
\begin{aligned}
f(x) &= \tfrac{1}{2}\left(\sum_{i=0}^{n-1} \lambda_i d_i\right)' A \left(\sum_{i=0}^{n-1} \lambda_i d_i\right) + b'\left(\sum_{i=0}^{n-1} \lambda_i d_i\right) + c \\
&= \sum_{i=0}^{n-1} \left(\tfrac{1}{2}\lambda_i^2 d_i' A d_i + \lambda_i b' d_i\right) + c
\end{aligned} \tag{3}
$$

Thus any quadratic can be minimized in n steps by minimizing the n terms independently.

Define $n \times i$ matrices Y_i and S_i

$$Y_i = [y_0, y_1, \ldots, y_{i-1}]$$
$$S_i = [s_0, s_1, \ldots, s_{i-1}]$$

Since $s_i = x_{i+1} - x_i$, $y_i = g_{i+1} - g_i$ and $g_i = Ax_i + b$ then,

$$y_i = As_i$$

and

$$Y_i' s_j = S_i' y_j = 0 \qquad 1 \le i \le j \le n-1 \tag{4}$$

when the steps $s_0, s_1, \ldots$ are conjugate.

Now consider two simple results that hold for independent directions d_i.

<u>Lemma 1.</u> The point $x_i = x_0 + \Sigma_{j=0}^{i-1} \alpha_j d_j$ is the minimum of $f(x)$ over the subspace $[d_0, d_1, \ldots, d_{i-1}]$ if and only if $S_i' g_i = 0$.

Proof: If $f(x_i)$ is a minimum in direction d_j then

$$[\partial f(x_i)]/\partial \alpha_j = d_j' g_i = 0 \qquad \text{for } 0 \le j < i-1, \qquad \text{i.e., } S_i' g_i = 0$$

Since $f(x)$ is strictly convex let $\hat{x}_i = x_i + \Sigma_{j=0}^{i-1} \epsilon_j d_j$; then $f(\hat{x}_i) \ge f(x_i) + g_i'(\hat{x}_i - x_i)$, equality occurring only when $\hat{x}_i = x_i$.

If $S_i' g_i = 0$, i.e., $d_j' g_i = 0$ for $0 \le j < i-1$; then $\epsilon_j \ne 0$ implies $f(\hat{x}_i) > f(x_i)$, and $f(x_i)$ is the minimum.

<u>Lemma 2.</u> Suppose at x_i, $S_i' g_i = 0$; if s_i satisfies $Y_i' s_i = 0$ and $s_i' g_{i+1} = 0$, then $S_{i+1} g_{i+1} = 0$.

Proof: On a quadratic function $Y_i' s_i = S_i' y_i = S_i'(g_{i+2} - g_{i+1}) = S_i' g_{i+2} = 0$. If in addition $s_i' g_{i+2} = 0$, then by definition of S_{i+1}, $S_{i+1}' g_{i+1} = 0$.

Lemma 1 provides a simple characterization of the progress at stage i, and lemma 2 indicates that stepping to a minimum in a direction orthogonal to the previous gradient changes locates the minimum over a larger subspace. Note that in neither case was conjugacy of the d_i required, only independence.

3. THE PROJECTED GRADIENT ALGORITHM

As Eq 4 shows, one way of generating conjugate directions is to make successive steps orthogonal to previous gradient changes. It is remarkable that this can be done on arbitrary functions $f(x)$ and produces a weaker form of conjugacy discussed elsewhere.[7] A result similar to Theorem 1 has been given independently by Goldfarb.[8]

Theorem 1

Let R be a symmetric positive definite matrix and define s_i, y_i by the recursion

$$\begin{aligned} i = 0 \quad & H_0 = I \\ i > 0 \quad & H_i = I - RY_i'(Y_i'RY_i)^{-1}Y_i' \end{aligned} \tag{5}$$

$$f(x_{i+1}) = \min_{\alpha_i} [f(x_i + \alpha_i H_i R g_i)] \tag{6}$$

$$S_{i+1} = [S_i, x_{i+1} - x_i] \tag{7}$$

$$Y_{i+1} = [Y_i, g_{i+1} - g_i] \tag{8}$$

then either for some $j < n$

$$H_j g_j = 0, \; H_j \neq 0 \text{ and } g_j = 0, \; x_j = x^*$$

or if the recursion continues to $j = n$

$$H_n = 0 \quad g_n = 0, \quad x_n = x^*$$

Proof: If $g_0 = 0$, then $H_0 = I$ and $x_0 = x^*$. If $g_0 \neq 0$, then $d_0 = H_0 R g_0 \neq 0$. Equation 6 requires $d_0' g_1 = 0$, i.e., $\alpha_1 = -d_0' g_0 / d_0 A d_0 < 0$, and consequently both s_0 and $y_0 \neq 0$. Thus $Y_1 = [y_0]$ and $S_1 = [s_0]$ both here rank 1, and $S_1' g_1 = s_0' g_1 = 0$.

Proceeding by induction, suppose Y_i and S_i have rank i and $S_i' g_i = 0$. Then H_i exists and is a projection matrix with properties $H_i^2 = H_i$, $H_i' Y_i = 0$. $H_i R Y_i = 0$. Thus from Eq 5 each new direction is orthogonal to the columns of Y_i.

Now $d_i = H_i R g_i = 0$ if and only if $g_i = 0$, for if $g_i \neq 0$, then by Eq 5 $g_i \in [y_0, y_1, \ldots, y_{i-1}]$, i.e., for some w, $g_i = Y_i w$. However, $S_i' g_i = 0$ implies $S_i' Y_i w = S_i' A S_i w = 0$ for $w \neq 0$, which contradicts the definiteness of A. Thus $H_i R g_i = 0$ implies $g_i = 0$.

Suppose $H_i R g_i \neq 0$; then Eq 6 requires $d_i' g_{i+1} = 0$, i.e., $\alpha_i = -g_i' R H_i' g_i / g_i' R H_i' A g_i < 0$ since g_i is not a linear combination of $y_0, y_1, \ldots, y_{i-1}$. Thus s_i

and y_i are nonzero. The direction choice implies $Y_i's_i = 0$ and as a result Y_{i+1}, S_{i+1} have rank $i + 1$. Otherwise for some $w \neq 0$, $y_i = Y_i w$ and $S_i'y_i = S_i' AS_i w = 0$ as before. Similarly if $s_i = S_i w \neq 0$, $Y_i's_i = S_i'AS_i w = 0$ implies $w = 0$.

The recursion terminates for some $j < n$ when $H_j R g_j = 0$, which requires $g_j = 0$ and $x_j = x^*$.

If the recursion continues to $i = n$, as is likely, then, since by induction Y_n, S_n have rank n and $S_n'g_n = 0$, it follows that $g_n = 0$, $x_n = x^*$, and $H_n = 0$.

A convenient algorithm can be found by application of the bordered inverse lemma in App A, to

$$H_i R = R - RY_i(Y_i'RY_i)^{-1}Y_i'R$$

Algorithm 1

$$\begin{aligned} H_{i+1} &= H_i - (H_i y_i)(H_i y_i)'/(y_i'H_i y_i) \\ H_0 &= R \end{aligned} \tag{9}$$

Then Eq 6 is replaced by

$$f(x_{i+1}) = \min_{\alpha_i}[f(x_i + \alpha_i H_i g_i)] \tag{10}$$

Corollary 1.1. If $Y_i's_i = 0$ for $i = 1, 2, \ldots, j$, then $s_0, s_1, \ldots, s_j$ are conjugate.

Proof: $Y_i's_i = 0$ implies $s_k'As_i = s_i'As_k = 0$ for $k < i$, i.e., $s_i'As_k = 0$ $0 \leq i \neq k \leq j$.

R allows a choice other than I for the initial H_0, a property that apparently minimizes round-off errors.[1] However, R can also be used to take advantage of any partial inverse of A.

Suppose A has a partitioned form

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

Assume A_{11}^{-1} is a known $r \times r$ block and set

$$R = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} = H_0 \tag{11}$$

Inserting this in Eq 10, a simple calculation shows that

$$\alpha_0 = -g_0'H_0g_0/g_0'H_0AH_0g_0 = -1$$

and that

$$g_1 = \begin{pmatrix} g_{01} \\ g_{02} \end{pmatrix} - A \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} g_{01} \\ g_{02} \end{pmatrix} = \begin{pmatrix} 0 \\ g_{02} - A_{21}A_{11}^{-1}g_{01} \end{pmatrix} \tag{12}$$

Thus the top r components of g_1 are zero after this "partial inverse" step.

Let $s_0, s_1, \ldots, s_{r-1}$ be r unit vectors of the form $s_i' = (0,0,\ldots,0,1,0,\ldots,0)$ with 1 in the i+1th position.

Let $Y_r = AS_r = (A_{11}', A_{21}')'$; then clearly S_r and Y_r have rank r, and if $g_r = g_1$ then $S_r'g_r = 0$. But these are the inductive hypotheses in Theorem 1 at the rth stage, which yields,

Corollary 1.2. After the partial inverse step (Eqs 10 and 11) the projected gradient algorithm is at stage r with H_r defined by

$$H_r = I - \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \left[\begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix}' \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \right]^{-1} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \tag{13}$$

It will terminate in not more than $n - r$ further steps.

A more transparent explanation of the restart is to note that if $A_{11}d_{i1} + A_{12}d_{i2} = 0$ where $d_i' = (d_{i1}', d_{i2}')$, then the top r components of g_i are unchanged from 0. This is equivalent to Eq 13. (See App C for an example.)

4. VARIABLE METRIC ALGORITHMS

The class of algorithms in this section are based on the following idea. If H_i satisfies $H_iY_i = S_i$ and steps $s_0, s_1, \ldots, s_{i-1}$ were obtained by minimizing down independent directions, i.e., $S_i'g_i = 0$, then the direction $d_i = H_i'g_i$ and step $s_i = \alpha_i d_i$ are conjugate to $s_0, s_1, \ldots, s_{i-1}$, i.e., $Y_i's_i = \alpha_i Y_i'H_i'g_i = \alpha_i S_i'g_i = 0$. Clearly if the process continues to stage n, $H_n = A^{-1}$, and all the steps are conjugate. Since g_n is orthogonal to the previous n steps it must be zero.

Now consider the general solution to the equations $H_iY_i = S_i$. This has the form for arbitrary Z of

$$H_i = S_iY_i^+ + Z(I - Y_iY_i^+) \tag{14}$$

where Y_i^+ is the generalized inverse of Y_i.[9] If Y_i is of rank i, then $Y_i^+ = (Y_i'Y_i)^{-1}Y_i'$ and has the property that $Y_iY_i^+$ is a projection of E^n onto $[y_0, y_1, \ldots, y_{i-1}]$. In addition $x^* = Y_i^+b$ minimizes $(Y_ix - b)'(Y_ix - b)$.

Suppose that Y_i has rank i; then it will be convenient to define $Y_i^* = (Y_i'HY_i)^{-1}\ Y_i'H$ where H is a positive definite symmetric matrix. Note that

$$H_i = S_i Y_i^* + R(I - Y_i Y_i^*)$$

also satisfies $H_i Y_i = S_i$, that $Y_i Y_i^* = Y_i(Y_i'H_i Y_i)\ Y_i'H$ also is a projection matrix, and that $x^* = Y_i^* b$ minimizes $(Y_i x - b)'H(Y_i x - b)$.

<u>Theorem 2, General Variable Metric Algorithm</u>

Let R and H be positive definite symmetric matrices and define the algorithm as follows:

for $i = 0$

$$H_0 = R \tag{15}$$

for $i > 0$

$$H_i = S_i Y_i^* + R(I - Y_i Y_i^{**}) \tag{16}$$

where

$$Y_i^* \text{ and } Y_i^{**} \text{ have the form } (Y_i'HY_i)^{-1} Y_i'H \tag{17}$$

with $H = R$ or A^{-1} independently for each term.

$$\begin{aligned} f(x_{i+1}) &= \min_{\alpha_i} f(x_i + \alpha_i H_i' g_i) \\ Y_{i+1} &= [Y_i \mid g_{i+1} - g_i] \end{aligned} \tag{18}$$

$$S_{i+1} = [S_i \mid x_{i+1} - x_i] \tag{19}$$

with

$$Y_1 = [g_1 - g_0],\ S_1 = [x_1 - x_0] \tag{20}$$

Then the algorithm terminates for some $i \le n$ when $H_i'g_i = 0$ implies $g_i = 0$, $x_i = x^*$. If $i = n$ then $H_n = A^{-1}$.

Proof: If $x_0 \neq x^*$ then $g_0 \neq 0$. Using Eqs 15, 18, and 20 the first step results in Y_1 and S_1 having rank 1 and $S_1'g_1 = 0$.

Proceeding by induction, suppose that at stage $i < n$, $g_i \neq 0$, Y_i and S_i have rank i and $S_i'g_i = 0$. It will be shown that this is true for $i + 1$.

Computing the direction of search, $H_i'g_i = Y_i^{*\prime} S_i' g_i + (I - Y_i^{**\prime} Y_i')\, Rg_i = 0 + [I - HY_i\, (Y_i'HY_i)^{-1}\, Y_i]\, Rg_i = 0$ if and only if $H^{-1}Rg_i \in [y_0, y_1, \ldots, y_{i-1}]$. But for $H = R$ or $H = A^{-1}$, $S_i'g_i = 0$ would require $g_i = 0$. Thus $H_i'g_i \neq 0$ if $g_i \neq 0$.

Minimizing at stage i, $\alpha_i = -g_i'H_i'g_i / g_i'H_i A H_i' g_i \neq 0$ if and only if $g_i'H_i'g_i > 0$. If $H = A^{-1}$ then $g_i'H_i'g_i = g_i'Rg_i > 0$. If $H = R$, $g_i'H_i'g_i = g_i'H_i R^{-1} H_i' g_i > 0$. Thus if $g_i \neq 0$, $\alpha_i \neq 0$. By construction, $Y_i'(x_{i+1} - x_i) = Y_i's_i = 0$. This fact, Eq 18, and lemma 2 provide that $S_{i+1}'g_{i+1} = 0$.

Since $\alpha_i \neq 0$, $x_{i+1} - x_i$ and $g_{i+1} - g_i$ are nonzero and S_{i+1}, Y_{i+1} in Eq 19 will have rank $i+1$. Otherwise $y_i \in [y_0, y_1, \ldots, y_{i-1}]$ and since $Y_i's_i = S_i'y_i = 0$, $y_i = 0$ and similarly $s_i = 0$, a contradiction.

Thus the iteration can only terminate if $g_i = 0$ for which $x_i = x_i^*$; otherwise it proceeds until $i = n$. Here, however, S_n has rank n and $S_n'g_n = 0$ implies $g_n = 0$ and $x_n = x^*$. $H_n = A^{-1}$ by construction since Y_n and S_n have rank n.

Since H can be chosen to be A^{-1} or R in Eqs 16 and 17 there are four possible algorithms in this scheme.

The next corollary allows for the fact that if a restart is used the initial directions $[s_0, s_1, \ldots, s_n]$ are not necessarily conjugate. See Corollaries 1.1 and 2.1. However under normal operations this is the case.

Corollary 2.1. If $Y_i's_i = 0$ for $i = 1, 2, \ldots, j < n$, then $s_0, s_1, \ldots, s_{j-1}$ are conjugate.

Particular algorithms can be obtained by choosing H differently for each Y_i^* in Eq 16.

Algorithm 2

Choose

$$H_i = S_i(S_i'Y_i)^{-1}S_i' + R(I - Y_i(S_i'Y_i)^{-1}S_i') \tag{21}$$

This corresponds to $H = A^{-1}$ in Eq 17. Expanding this formula using the bordered inverse lemma in App A,

$$\begin{aligned} H_{i+1} &= H_i + (s_i - H_i y_i)(s_i - S_i \Delta Y_i' s_i)' / y_i'(I - S_i \Delta Y_i') s_i \\ H_0 &= R \end{aligned} \tag{22}$$

where

$$\Delta = S_i'Y_i$$

When used in Theorem 2, the projection properties of H_i require that $Y_i's_i = 0$ and consequently

$$\begin{aligned} H_{i+1} &= H_i + (s_i - H_i y_i)(s_i)' / s_i' y_i \\ H_0 &= R \end{aligned} \tag{23}$$

This particular algorithm is due independently to G. P. McCormick. Note that in general H_i is unsymmetric.

Algorithm 3

Choose

$$H_i = S_i(Y_i'RY_i)^{-1}Y_i'R + R[I - Y_i(Y_i'RY_i)^{-1}Y_i'R] \tag{24}$$

This corresponds to $H = R$ throughout Theorem 2. Expanding this formula and using the projection properties of H_i in the form $Y_i's_i = 0$,

$$\begin{aligned} H_{i+1} &= H_i + (s_i - H_iy_i)(H_i'y_i)'/y_i'H_iy_i \\ H_0 &= R \end{aligned} \tag{25}$$

Again H_i will be unsymmetric and in particular here $H_i'y_i \neq H_iy_i$.

Algorithm 4

Choose

$$H_i = S_i(S_i'Y_i)^{-1}S_i' + R[I - Y_i(Y_i'RY_i)^{-1}Y_i'R] \tag{26}$$

This corresponds to $H = A^{-1}$ in the first Y_i^* and $H = R$ in the second Y_i^* of Eq 16. Expanding H_i and using the projection properties of H_i in the form $Y_i's_i = S_i'y_i = 0$,

$$\begin{aligned} H_{i+1} &= H_i - (H_iy_i)(H_iy_i)'/y_i'H_iy_i + s_is_i'/s_i'y_i \\ H_0 &= R \end{aligned} \tag{27}$$

This is immediately recognized as Fletcher and Powell's modification of Davidon's algorithm.[5] H_i is symmetric and the search direction $H_i'g_i = H_ig_i$ as commonly used.

As for the projection matrix, let s_i, $i = 0, 1, \ldots, r-1$ be the first r unit vectors and define $Y_r = AS_r$. Then if $g_r = g_1$ given by Eq 12, $S_r'g_r = 0$ and both S_r, Y_r have rank r. A simple calculation shows that for the Fletcher-Powell algorithm

$$S_rY_r^* = S_r(S_r'Y_r)^{-1}S_r' = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} \tag{28}$$

<u>Corollary 2.2.</u> After a partial inverse step (Eqs 18 and 11) the Fletcher-Powell-Davidon algorithm can be restarted from the rth stage with

$$H_r = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \left\{ I - \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \left[\begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix}' \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \right]^{-1} \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix}' \right\} \tag{29}$$

It will terminate in not more than $n - r$ further steps. An example is given in App C.

In principle the general variable metric method can be considered. All that is required is that independent directions $s_0, s_1, \ldots, s_{r-1}, y_0, y_1, \ldots, y_{r-1}$ be found such that for some initial estimate $H_0 = R$, $S_r Y_r^* = R$, and $S_r' g_r = 0$.

A fifth algorithm can be derived analogous to Algorithm 4 by inserting $H = A$ in the first Y_i^* of Eq 16 and A^{-1} in the second. Unfortunately it does not lead to a readily computable formula as do the others.

5. NUMERICAL RESULTS

Results of testing these algorithms on nonquadratic functions will now be given. The numerical procedure for the seven schemes considered is as follows.

Given $f(x)$, $g(x)$, and possibly $A(x)$, the matrix of second partial derivatives of $f(x)$ evaluated at x, and starting at x_0 with $H_0 = R$, the initial matrix, and normalizing d_i,

(a) Find the first local minimum of $f(x_i + \alpha_i d_i)$

$$f(x_{i+1}) = \min_{\alpha_i} f(x_i + \alpha_i d_i) \tag{30}$$

$$d_i = H_i' g_i / g_i' H_i H_i' g_i \tag{31}$$

(b) Update H_i according to the algorithm used.

<u>Algorithm 1, Projected Gradient Method (P-G)</u>

$$H_{i+1} = H_i - (H_i y_i)(H_i y_i)'/(y_i' H_i y_i) \tag{32}$$

for $i = n$, and every n steps $H_i = R$.

<u>Algorithm 2</u>

$$H_{i+1} = H_i + (s_i - H_i y_i) s_i' / s_i' y_i \tag{33}$$

<u>Algorithm 3</u>

$$H_{i+1} = H_i + (s_i - H_i y_i)(H_i' y_i)'/(y_i' H_i y_i) \tag{34}$$

Algorithm 4, Fletcher-Powell-Davidon (F-P-D)

$$H_{i+1} = H_i - (H_i y_i)(H_i y_i)'/(y_i' H_i y_i) + s_i s_i'/y_i' s_i \tag{35}$$

Algorithm 5, Newton-Raphson (N-R)

$$H_i = [A(x_i)]^{-1} \tag{36}$$

The program uses a modified Newton-Raphson step when it appears that $A(x_i)$ has negative eigenvalues as identified during the process of inversion of $A(x_i)$ using the Crout procedure. In this case the direction of move is along an eigenvector corresponding to a negative eigenvalue. By this means a region is located where the function is convex.[10]

Algorithm 6, Fletcher-Reeves (F-R)

$$\begin{aligned} d_0 &= -g_0 \\ d_{i+1} &= -g_{i+1} + d_i (g_{i+1}' g_{i+1} / g_i' g_i) \end{aligned} \tag{37}$$

for $i = n + 1$, and every $n + 1$ steps $d_i = -g_i$.

Algorithm 7, Projected Newton-Raphson (P-N-R)

$$H_{i+1} = H_i - (H_i y_i)(H_i y_i)'/(y_i' H_i y_i) \tag{38}$$

$$R_{i+1} = R_i + (s_i - R_i y_i)(H_i y_i)'/(y_i' H_i y_i) \tag{39}$$

for $i = n$, and every n steps $H_i = R_i$.

The last method investigates the effect of solving $R_i Y_i = S_i$ exactly using the schemes of Sec 4 in the absence of quadraticity. $H_i y_i$ provides the projection of y_i orthogonal to $[y_0, y_1, \ldots, y_{i-1}]$. Every n steps R_i is an approximation to $A(x_i)^{-1}$ and a Newton-Raphson move is made.

The reset form of the algorithm is obtained by resetting H_{n+1} for Algorithms 2, 3, 4, and 7 to R and restarting. Algorithm 1 must be reset every n steps and, in Algorithm 6, d_{n+1} is reset to $-g_{n+1}$ always.

The linear minimization is performed by a Fibonacci search. Cubic interpolation works well on low-order polynomial functions but does not prove adequate for he logarithmic penalty functions used in the Sequential Unconstrained Minimization Technique (SUMT) for which Algorithms 1 to 7, plus several others, make up an experimental XMOVE subroutine.[11,12]

Five problems were considered. The data for these, and other information, are found in App B. The numerical results are of course strictly comparative for each problem. In each case the fastest algorithm is indicated by encircled and italicized iteration numbers.

Table 1 gives results for Rosenbrock's banana-shaped valley.[13]

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{40}$$

Starting point $(x_1, x_2) = (-1.2, 1.0)$; the numbers quoted are iterations until $f(x^*) < 10^{-13}$.

TABLE 1

Numerical Results of Problem 1

Algorithm	Mode	
	Normal	Reset
1, P-G	—	42
2	18	31
3	21	37
4, F-P-D	19	35
5, N-R	*12*	—
6, F-R	—	*16*
7, P-N-R	36	21

TABLE 2

Numerical Results of Problem 2

Algorithm	Normal	Reset
1, P-G	—	65
2	36	47
3	46	47
4, F-P-D	40	49
5, N-R	*23*	—
6, F-R	—	*30*
7, P-N-R	58	55

Table 2 gives results for a test function credited to C. F. Wood of Westinghouse Research Laboratory:

$$\begin{aligned} f(x) = {} & 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ & + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1) \end{aligned} \tag{41}$$

This is designed to have a nonoptimal stationary point that can cause premature convergence. Initial point $(x_1, x_2, x_3, x_4) = (-3, -1, -3, -1)$, and the number of iterations is for $f(x^*) < 10^{-13}$.

Table 3 shows the results for a test problem formulated by the Shell Development Company,

$$f(x) = \sum_{j=1}^{5} c_j x_j + \sum_{j=1}^{5} \sum_{i=1}^{5} c_{ij} x_i x_j + \sum_{j=1}^{5} d_j x_j^3$$

subject to

$$x_j \geq 0, \quad j = 1,2,\ldots,5$$

$$\sum_{j=1}^{5} a_{ij}x_j \geq b_i, \quad i = 1,2,\ldots,10 \qquad (42)$$

This is a linearly constrained problem, which for particular choices of e_j, c_{ij}, d_j has a convex objective.[12] For this problem, SUMT replaces $f(x)$ by $f(x) - r \sum_{i=1}^{10} \log_e g_i(x)$ for a parameter $r > 0$, where $g_i(x) \geq 0$ represents the ith inequality constraint. If $x^*(r)$ is the solution of the modified problem, then $x^*(r) \to x^*$ as $r \to 0$ where x^* is the solution to Eq 42.

TABLE 3

Numerical Results of Problem 3

Algorithm	$r = 1.0$		$r = 1.56 \times 10^{-2}$		$r = 2.44 \times 10^{-4}$	
	Normal	Reset	Normal	Reset	Normal	Reset
1, P-G	—	26	—	55	—	70
2	27	22	44	41	62	60
3	33	22	50	40	67	54
4, F-P-D	27	22	46	40	60	56
5, N-R	11	—	17	—	22	—
6, F-R	—	34	—	> 165	—	Fail
7, P-N-R	31	20	50	37	67	54

Table 4 gives results for the dual to the previous problem. Here the dual problem has a cubic objective and quadratic constraints.[10]

TABLE 4

Numerical Results of Problem 4

Algorithm	$r = 1.0$		$r = 0.25$		$r = 0.0625$	
	Normal	Reset	Normal	Reset	Normal	Reset
1, P-G	—	120	—	169	—	211
2	134	98	195	132	221	187
3	136	100	220	134	246	168
4, F-P-D	406	97	473	133	500	169
5, N-R	30	—	16	—	48	—
6, F-R	—	> 489	—	Fail	—	Fail
7, P-N-R	166	113	198	150	230	186

Finally, Table 5 shows the results for an intriguing problem of maximizing the area of a hexagon subject to the constraint that its maximum diameter is 1. It is interesting to note that the solution is <u>not</u> a regular hexagon.[14] The particular formulation used had 9 variables and 13 inequality constraints, although there is a certain amount of redundancy.

TABLE 5

Numerical Results of Problem 5

Algorithm	$r = 1.0$		$r = 10^{-2}$		$r = 10^{-4}$		$r = 10^{-6}$	
	Normal	Reset	Normal	Reset	Normal	Reset	Normal	Reset
1, P-G	—	13	—	55	—	194	—	278
2	17	20	34	42	308	79	326	97
3	11	11	31	32	73	56	96	70
4, F-P-D	47	18	66	40	206	64	215	80
5, N-R	18	—	30	—	54	—	65	—
6, F-R	—	13	—	55	—	194	—	278
7, P-N-R	19	23	51	58	92	91	120	101

Summary of Results

Tables 1 to 5 illustrate primarily the difficulty of selecting a meaningful test problem. The first two problems are smooth polynomials albeit with odd-shaped valleys. The next three problems are basically quadratics or cubics with infinite barriers of the penalty functions against which the solutions lie. This means that their hessian matrices $A(x_i)$ become very ill-conditioned, for the binding constraints correspond to large eigenvalues, which tend to infinity for fixed r as the solution $x_i(r)$ approaches the constraint.

If second derivatives are available, the Newton-Raphson method is clearly the best for all five problems.

It seems that on smooth polynomials the variable metric methods are best <u>not</u> reset, while on penalty functions they are best reset. Under these conditions, the Fletcher-Powell-Davidon algorithm is better for the former class of problems and Algorithm 3 is better for the latter class, the penalty functions. It is remarkable that in Table 4 Algorithms 2, 3, and in particular 4 were extremely slow when not reset.

Finally Algorithm 7, the projected Newton-Raphson, is better than the projected gradient, showing that second-order information helps. However, the separate calculation to obtain R_N exactly (Eq 39) does not seem to merit the effort compared with the other schemes.

6. CONCLUSIONS

This paper has unified a series of algorithms in a single framework. Basically, this is that variable metric schemes depend on the generalized solution to a set of linear equations, and their associated projection properties give rise to conjugate directions. A result of this general approach has been three new algorithms whose comparative numerical properties are promising. Extensions to this work will be found elsewhere.[7]

APPENDIXES

Appendix A

BORDERED INVERSE LEMMA

As an application of the matrix inverse lemma consider

$$H_i = A(B'C)^{-1}D'$$

where A, B, C, D are all $n \times i$ matrices with $i < n$ such that they have rank i.

Let a, b, c, d be n-vectors such that $[A, a]$, etc, have rank $i + 1$ and consider

$$\begin{aligned} H_{i+1} &= [A, a]([B, b]'[C, c])^{-1}[D, d]' \\ &= [A, a]\begin{bmatrix} B'C & B'c \\ b'C & b'c \end{bmatrix}^{-1}[D, d]' \end{aligned}$$

Applying the bordered inverse lemma to the center matrix,[18]

$$H_{i+1} = [A, a]\left(\begin{bmatrix} (B'C)^{-1} & 0 \\ 0 & 0 \end{bmatrix} + [-c'BR', 1]'\Delta^{-1}[-b'CR, 1]\right)[D, d]'$$

where

$$\begin{aligned} R &= (B'C)^{-1} \\ \Delta &= b'(I - CRB')C \end{aligned}$$

Now multiplying through yields,

$$\begin{aligned} H_{i+1} &= A(B'C)^{-1}D' + (ARB'c + a)\Delta^{-1}(-b'CRD' + d') \\ &= H_i + \frac{(a - A(B'C)^{-1}B'c)(d - D(C'B)^{-1}C'b)'}{b'(I - C(B'C)^{-1}B')c} \end{aligned}$$

giving the basic formula used throughout this work.

Appendix B

TEST PROBLEM DATA

Problem 1. Rosenbrock's Banana-Shaped Valley[13]

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Problem 2. Wood's Function

$$f(x) = 100(x_2 - x_1^2) + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2$$
$$+ 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$$

Problem 3. Shell Primal Problem

$$f(x) = \sum_{j=1}^{j=5} c_j x_j + \sum_{i=1}^{i=5} \sum_{j=1}^{j=5} c_{ij} x_i x_j + \sum_{j=1}^{5} d_j x_j^3$$

subject to $x_j \geq 0 \quad j = 1, 2, \ldots, 5$

$$\sum_{j=1}^{j=5} a_{ij} x_j \geq b_i \qquad i = 1, 2, \ldots, 10$$

The data for c_j**,** c_{ij}**,** d_j**,** a_{ij}**, and** b_i **are given in Table B1.**

Problem 4. Shell Dual Problem

Maximize

$$f(x) = \sum_{j=1}^{j=10} b_j y_j - \sum_{i=1}^{i=5} \sum_{j=1}^{j=5} c_{ij} x_i x_j - 2 \sum_{i=1}^{5} d_i x_i^3$$

subject to

$$\sum_{j=1}^{10} a_{ji} y_j \leq c_i + 2 \sum_{j=1}^{5} c_{ji} x_j + 3 d_i x_i^2, \quad i = 1, 2, \ldots 5$$
$$x_i \geq 0 \quad i = 1, 2, \ldots, 5$$
$$y_i \geq 0 \quad i = 1, 2, \ldots, 10$$

TABLE B1

Data for Problems 3 and 4

($n = 5$)

		1	2	3	4	5			Other b's
e_j		−15	−27	−36	−18	−12			
c_{ij}	1	30	−20	−10	32	−10			
	2	−20	39	−6	−31	32			
	3	−10	−6	10	−6	−10			
	4	32	−31	−6	39	−20			
	5	−10	32	−10	−20	30			
d_j		4	8	10	6	2	b_i	b_i'	b_i''
a_{ij}	1	−16	2	0	1	0	−40	−40	−40
	2	0	−2	0	0.4	2	−2	−2	−2
	3	−3.5	0	2	0	0	−0.25	−0.5	−1
	4	0	−2	0	−4	−1	−4	−4	−4
	5	0	−9	−2	1	−2.8	−4	−8	−16
	6	2	0	−4	0	0	−1	−2	−4
	7	−1	−1	−1	−1	−1	−40	−40	−40
	8	−1	−2	−3	−2	−1	−60	−60	−60
	9	1	2	3	4	5	5	2.5	2.5
	10	1	1	1	1	1	1	1	1

Problem 5. Hexagon Problem

Maximize $f(x)$ the area of a hexagon where,

$$f(x) = \tfrac{1}{2}\left|x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7\right|$$

subject to constraints that the maximum diameter is unity

$$1 \geq x_3^2 + x_4^2$$

$$1 \geq x_9^2$$

$$1 \geq (x_5^2 + x_6^2)$$

$$1 \geq x_1^2 + (x_2 - x_9)^2$$

$$1 \geq (x_1 - x_5)^2 + (x_2 - x_6)^2$$

$$1 \geq (x_1 - x_7)^2 + (x_2 - x_8)^2$$

$$1 \geq (x_3 - x_5)^2 + (x_4 - x_6)^2$$

$$1 \geq (x_3 - x_7)^2 + (x_4 - x_8)^2$$

$$1 \geq x_7^2 + (x_8 - x_9)^2$$

and that the figure described is a nondegenerate hexagon

$$x_1x_4 - x_2x_3 \geq 0$$

$$x_3x_9 \geq 0$$

$$x_5x_9 \geq 0$$

$$x_5x_8 - x_6x_7 \geq 0$$

$$x_9 \geq 0$$

The figure described has one diameter on the vertical axis. The problem is not expressed in its simplest form.

Appendix C

EXAMPLES OF RESTARTS

Example of Restart after a Partial Inverse Move for the Projected Gradient Scheme

Suppose $n = 2$ and A and b are

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

A partial inverse of the leading 2×2 submatrix of A has the form

$$H = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} = H_0$$

This is used as the starting matrix.

Iteration 0, the initial point

x_0	g_0	y_0	s_0
10	31	.	.
10	31	.	.
10	41	.	.

Iteration 1, after the partial inverse step

x_1	g_1	y_0	s_0
10	0	31	0
-21	0	-31	-31
10	10	-31	0

H is now reset to

$$H_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \left[\begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1/6 & -1/3 & 1/6 \\ -1/3 & 2/3 & -1/3 \\ 1/6 & -1/3 & 1/6 \end{pmatrix}$$

Iteration 2, after one projected gradient stage

x_2	g_2	y_2	s_2
0	0	0	-10
-1	0	0	20
0	0	-10	-10

$$H_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

after updating using $H_2 y_2$ above.

To check, $g = Ax^* + b$ should be zero.

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

This completes the problem after two steps instead of the normal three.

Example of a Partial Inverse Move for the Fletcher-Powell-Davidon Scheme

Suppose for $n = 2$, A and b are given by

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

A partial inverse of the leading 2×2 submatrix gives

$$R = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} = H_0$$

Start, iteration 0

x_0	g_0	y_0	s_0
10	31	.	.
10	31	.	.
10	41	.	.

Iteration 1, after the partial inverse step

x_1	g_1	y_0
10	0	-31
-21	0	-31
10	10	-31

H_2 is now reset to

$$H_2 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \left[\begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 7/6 & -4/3 & 1/6 \\ -4/3 & 8/6 & -1/3 \\ 1/6 & -1/3 & 1/6 \end{pmatrix}$$

Iteration 2

x_2	g_2	y_2	s_2
0	0	0	-10
-1	0	0	20
0	0	-10	-10

H_3 after updating using $H_2 y_2$ and s_2 is

$$H_3 = \begin{pmatrix} 2 & -3 & 1 \\ -3 & 6 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Check on the inverse

$$AH_3 = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{pmatrix} 2 & -3 & 1 \\ -3 & 6 & -2 \\ 1 & -2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This completes the problem after two steps instead of the normal three.

REFERENCES

1. W. C. Davidon, "Variable Metric Method for Minimization," Research and Development Rept ANL-5990 (rev), US Atomic Energy Commission, Argonne National Laboratories, 1959.
2. C. G. Broyden, "Quasi-Newton Methods and Their Application to Function Minimization," Math. of Computation, 21(99): 368–81 (1967).
3. A. V. Fiacco and G. P. McCormick, Sequential Unconstrained Minimization Techniques for Nonlinear Programming, John Wiley & Sons, Inc., New York, to be published in 1968.
4. P. Wolfe, "Methods of Nonlinear Programming," in Graves and Wolfe (eds), Recent Advances in Mathematical Programming, McGraw-Hill Book Co., New York, pp 67–86, 1963.
5. R. Fletcher and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," Computer J., 6: 163–68 (1963).
6. M. R. Hestenes, "The Conjugate Gradient Method for Solving Linear Systems," Proceedings of the Symposium on Applied Mathematics, Vol VI, pp 83–102, McGraw-Hill Book Co., New York, 1956.
7. G. P. McCormick and J. D. Pearson, "Variable Metric Methods, Penalty Functions, and Unconstrained Optimization," submitted to Conference on Optimization, Keele Hall, Staffordshire, England, 1968.
8. D. Goldfarb, "Extension of Davidon's Variable Metric Method to Maximization Under Linear Inequality and Equality Constraints," presented at SIAM National Meeting, Iowa City, Iowa, May 66.
9. A. Ben-Israel and A. Charnes, "Contributions to the Theory of Generalized Inverses," SIAM J. Appl. Math., 11(3): 667–99 (1963).
10. G. P. McCormick and W. I. Zangwill, "A Technique for Calculating Second-Order Optima," technical paper in preparation, Research Analysis Corporation, McLean, Virginia.
11. A. V. Fiacco and G. P. McCormick, "Computational Algorithm for the Sequential Unconstrained Minimization Technique for Nonlinear Programming," Mgt. Sci., 10(4): 601–17 (1964).
12. G. P. McCormick, W. C. Mylander, and A. V. Fiacco, "Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming," RAC-TP-151, Research Analysis Corporation, Apr 65.
13. H. H. Rosenbrock, "Automatic Method for Finding the Greatest or Least Values of a Function," Computer J., (3): 175–84 (1960).
14. W. C. Mylander, "A Geometric Problem Solved by Nonlinear Programming," technical paper in preparation, Research Analysis Corporation.
15. J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming—Part 1, Linear Constraints," SIAM J. Appl. Math., (9): 414–43 (1961).

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Research Analysis Corporation McLean, Virginia 22101	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

ON VARIABLE METRIC METHODS OF MINIMIZATION

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Paper

5. AUTHOR(S) (First name, middle initial, last name)

John D. Pearson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1968	33	15

8a. CONTRACT OR GRANT NO. DA-44-188-ARO-1 b. PROJECT NO. 008.112 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) RAC-TP-302 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Army Research Office, Office of the Chief of Research and Development

13. ABSTRACT

Two basic approaches to the generation of conjugate directions are considered for the problem of unconstrained minimization of a quadratic function. Using the principle of choosing a step direction orthogonal to the previous gradient changes, a projected gradient algorithm and a class of variable metric algorithms are derived. Three variants of the class are developed into algorithms, one of which is the Fletcher-Powell-Davidon scheme.

Numerical results indicate the merits of the new algorithms compared to several now in use, for a variety of nonquadratic problems.

DD FORM 1 NOV 65 1473

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
variable metric algorithm						
projected gradient algorithm						
Fletcher-Powell-Davidon algorithm						
first-order derivative algorithm						
quasi-Newton algorithm						